# Red Hat OpenShift AI Self-Managed 2-latest

## Upgrading OpenShift AI Self-Managed

Upgrade OpenShift AI on OpenShift Container Platform

Last Updated: 2024-04-05

# Red Hat OpenShift AI Self-Managed 2-latest Upgrading OpenShift AI Self-Managed

Upgrade OpenShift AI on OpenShift Container Platform

## Legal Notice



## Abstract

Upgrade OpenShift AI on OpenShift Container Platform.

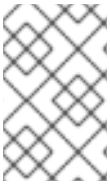
# Table of Contents

# PREFACE

As a cluster administrator, you can configure either automatic or manual upgrade of the OpenShift AI Operator.

# CHAPTER 1. OVERVIEW OF UPGRADING OPENSHIFT AI SELF-MANAGED

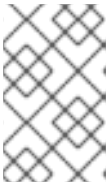
As a cluster administrator, you can configure either automatic or manual upgrades for the Red Hat OpenShift AI Operator.



**NOTE**

For information about upgrading OpenShift AI as self-managed software on your OpenShift cluster in a disconnected environment, see Upgrading OpenShift AI Self-Managed in a disconnected environment.

- If you configure automatic upgrades, when a new version of the Red Hat OpenShift AI Operator is available, Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without human intervention.
- If you configure manual upgrades, when a new version of the Red Hat OpenShift AI Operator is available, OLM creates an update request.
  A cluster administrator must manually approve the update request to update the Operator to the new version. See Manually approving a pending Operator upgrade for more information about approving a pending Operator upgrade.
- By default, the Red Hat OpenShift AI Operator follows a sequential update process. This means that if there are several minor versions between the current version and the version that you plan to upgrade to, Operator Lifecycle Manager (OLM) upgrades the Operator to each of the minor versions before it upgrades it to the final, target version. If you configure automatic upgrades, OLM automatically upgrades the Operator to the latest available version, without human intervention. If you configure manual upgrades, a cluster administrator must manually approve each sequential update between the current version and the final, target version.
  Red Hat supports the current version and three previous minor versions of OpenShift AI Self-Managed. For more information, see the Red Hat OpenShift AI Self-Managed Life Cycle knowledgebase article.
- When you upgrade OpenShift AI, you should complete the Requirements for upgrading OpenShift AI.
- If you upgrade to OpenShift AI from version 1 (OpenShift Data Science), follow the guidelines in Cleaning up unused resources from version 1 of Red Hat OpenShift AI (OpenShift Data Science).
- Before you can use an accelerator in OpenShift AI, your instance must have the associated accelerator profile. If your OpenShift Container Platform instance has an accelerator, its accelerator profile is preserved after an upgrade. For more information about accelerators, see Working with accelerators.
- Notebook images are integrated into the image stream during the upgrade and subsequently appear in the OpenShift AI dashboard.



**NOTE**

Notebook images are constructed externally; they are prebuilt images that undergo quarterly changes and they do not change with every OpenShift AI upgrade.

### Additional resources

- Operator Lifecycle Manager workflow

# CHAPTER 2. CONFIGURING THE UPGRADE STRATEGY FOR OPENSHIFT AI

As a cluster administrator, you can configure either an automatic or manual upgrade strategy for the Red Hat OpenShift AI Operator.

**IMPORTANT**

By default, the Red Hat OpenShift AI Operator follows a sequential update process. This means that if there are several versions between the current version and the version that you intend to upgrade to, Operator Lifecycle Manager (OLM) upgrades the Operator to each of the intermediate versions before it upgrades it to the final, target version. If you configure automatic upgrades, OLM automatically upgrades the Operator to the *latest* available version, without human intervention. If you configure manual upgrades, a cluster administrator must manually approve each sequential update between the current version and the final, target version.

For information about supported versions, see Red Hat OpenShift AI Life Cycle.

**Prerequisites**

- You have cluster administrator privileges for your OpenShift Container Platform cluster.
- The Red Hat OpenShift AI Operator is installed.

**Procedure**

1. Log in to the OpenShift Container Platform cluster web console as a cluster administrator.
2. In the **Administrator** perspective, in the left menu, select **Operators → Installed Operators**.
3. Click the **Red Hat OpenShift AI** Operator.
4. Click the **Subscription** tab.
5. Under **Update approval**, click the pencil icon and select one of the following update strategies:
   - **Automatic**: New updates are installed as soon as they become available.
   - **Manual**: A cluster administrator must approve any new update before installation begins.
6. Click **Save**.

**Additional resources**

- For more information about the subscription channels that are available in version 2 of the Red Hat OpenShift AI Operator, see Installing the Red Hat OpenShift AI Operator.
- For more information about upgrading Operators that have been installed by using OLM, see Updating installed Operators in the OpenShift Container Platform documentation.

# CHAPTER 3. REQUIREMENTS FOR UPGRADING OPENSHIFT AI

This section describes the tasks that you should complete when upgrading OpenShift AI.

**Check the components in the DataScienceCluster object**

When you upgrade Red Hat OpenShift AI, the upgrade process automatically uses the values from the previous **DataScienceCluster** object.

After the upgrade, you should inspect the **DataScienceCluster** object and optionally update the status of any components as described in Updating the installation status of Red Hat OpenShift AI components by using the web console.

**Recreate existing pipeline runs**

When you upgrade to a newer version, any existing pipeline runs that you created in the previous version continue to refer to the image for the previous version (as expected).

You must delete the pipeline runs (not the pipelines) and create new pipeline runs. The pipeline runs that you create in the newer version correctly refer to the image for the newer version.

For more information on pipeline runs, see Managing pipeline runs.

**Address KServe requirements**

For KServe (single-model serving platform), you must meet these requirements:

- Install dependent Operators, including the Red Hat OpenShift Serverless and Red Hat OpenShift Service Mesh Operators. For more information, see Serving large models.
- After the upgrade, you must inspect the default **DataScienceCluster** object and verify that the value of the **managementState** field for the **kserve** component is **Managed**.
- In Red Hat OpenShift AI version 2.4, the KServe component is a Limited Availability feature. If you enabled the **kserve** component and created models in version 2.4, then after you upgrade to version 2.5, you must update some OpenShift AI resources as follows:

  1. Log in as an administrator to the OpenShift Container Platform cluster where OpenShift AI 2.5 is installed:

     ```
     $ oc login
     ```

  2. Update the **DSC Initialization** resource:

     ```
     $ oc patch $(oc get dsci -A -oname) --type='json' -p='[{"op": "replace", "path": "/spec/serviceMesh/managementState", "value":"Unmanaged"}]'
     ```

  3. Update the **Data Science Cluster** resource:

     ```
     $ oc patch $(oc get dsc -A -oname) --type='json' -p='[{"op": "replace", "path": "/spec/components/kserve/serving/managementState", "value":"Unmanaged"}]'
     ```

  4. Update the **InferenceServices** CRD:

```
$ oc patch crd inferenceservices.serving.kserve.io --type=json -p='[{"op": "remove",
"path": "/spec/conversion"}]'
```

5. Optionally, restart the Operator pod.
   For more information about these configurations, see Installing KServe.

- If you deployed a model by using KServe in OpenShift AI version 2.4, when you upgrade to version 2.5 the model does not automatically appear in the OpenShift AI dashboard. To update the dashboard view, redeploy the model by using the OpenShift AI dashboard.

# CHAPTER 4. CLEANING UP UNUSED RESOURCES FROM VERSION 1 OF RED HAT OPENSHIFT AI (OPENSHIFT DATA SCIENCE)

Version 1 of OpenShift AI (previously OpenShift Data Science) created a set of Kubeflow Deployment Definition (that is, **KfDef**) custom resource instances on your OpenShift Container Platform cluster for various components of OpenShift AI. When you upgrade to version 2, these resources are no longer used and require manual removal from your cluster. The following procedures shows how to remove unused **KfDef** instances from your cluster by using both the OpenShift command-line interface (CLI) and the web console.

## 4.1. REMOVING UNUSED RESOURCES BY USING THE CLI

The following procedure shows how to remove unused **KfDef** instances from the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects in your OpenShift Container Platform cluster by using the OpenShift command-line interface (CLI). These resources become unused after you upgrade from version 1 to version 2 of OpenShift AI.

### Prerequisites

- You upgraded from version 1 to version 2 of OpenShift AI.
- You have cluster administrator privileges for your OpenShift Container Platform cluster.

### Procedure

1. Open a new terminal window.
2. In the OpenShift command-line interface (CLI), log in to your on your OpenShift Container Platform cluster as a cluster administrator, as shown in the following example:

   ```
   $ oc login <openshift_cluster_url> -u system:admin
   ```

3. Delete any **KfDef** instances that exist in the **redhat-ods-applications** project.

   ```
   $ oc delete kfdef --all -n redhat-ods-applications --ignore-not-found || true
   ```

   For any **KfDef** instance that is deleted, the output is similar to the following example:

   ```
   kfdef.kfdef.apps.kubeflow.org "rhods-dashboard" deleted
   ```

   **TIP**

   If deletion of a **KfDef** instance fails to finish, you can force deletion of the object using the information in the "Force individual object removal when it has finalizers" section of the following Red Hat solution article: https://access.redhat.com/solutions/4165791.

4. Delete any **KfDef** instances in the **redhat-ods-monitoring** and **rhods-notebooks** projects by entering the following commands:

```
$ oc delete kfdef --all -n redhat-ods-monitoring --ignore-not-found || true
$ oc delete kfdef --all -n rhods-notebooks --ignore-not-found || true
```

**Verification**

- Check whether all **KfDef** instances have been removed from the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects.

  ```
  $ oc get kfdef --all-namespaces
  ```

  Verify that you see no **KfDef** instances listed in the **redhat-ods-applications**, **redhat-ods-monitoring**, or **rhods-notebooks** projects.

## 4.2. REMOVING UNUSED RESOURCES BY USING THE WEB CONSOLE

The following procedure shows how to remove unused **KfDef** instances from the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects in your OpenShift Container Platform cluster by using the OpenShift web console. These resources become unused after you upgrade from version 1 to version 2 of OpenShift AI.

**Prerequisites**

- You upgraded from version 1 to version 2 of OpenShift AI.
- You have cluster administrator privileges for your OpenShift Container Platform cluster.

**Procedure**

1. Log in to the OpenShift Container Platform web console as a cluster administrator.
2. In the web console, click **Administration → CustomResourceDefinitions**.
3. On the **CustomResourceDefinitions** page, click the **KfDef** custom resource definition (CRD).
4. Click the **Instances** tab.
   The page shows all **KfDef** instances on the cluster.
5. Take note of any **KfDef** instances that exist in the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects. These are the projects that you will clean up in the remainder of this procedure.
6. To delete a **KfDef** instance from the **redhat-ods-applications**, **redhat-ods-monitoring**, or **rhods-notebooks** project, click the action menu (⋮) beside the instance and select **Delete KfDef** from the list.
7. To confirm deletion of the instance, click **Delete**.

   **TIP**

   If deletion of a **KfDef** instance fails to finish, you can force deletion of the object using the information in the "Force individual object removal when it has finalizers" section of the following Red Hat solution article: https://access.redhat.com/solutions/4165791.

8. Repeat the preceding steps to delete all remaining **KfDef** instances that you see in the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects.

# CHAPTER 5. ADDING A CA BUNDLE AFTER UPGRADING

Red Hat OpenShift AI 2-latest provides support for using self-signed certificates. If you have upgraded from OpenShift AI 2.7 or earlier versions, you can add self-signed certificates to the OpenShift AI deployments and Data Science Projects in your cluster.

There are two ways to add a Certificate Authority (CA) bundle to OpenShift AI. You can use one or both of these methods:

- For OpenShift Container Platform clusters that rely on self-signed certificates, you can add those self-signed certificates to a cluster-wide Certificate Authority (CA) bundle (**ca-bundle.crt**) and use the CA bundle in Red Hat OpenShift AI.
- You can use self-signed certificates in a custom CA bundle (**odh-ca-bundle.crt**) that is separate from the cluster-wide bundle.

For more information, see Working with certificates.

### Prerequisites

- You have admin access to the **DSCInitialization** resources in the OpenShift Container Platform cluster.
- You installed the OpenShift command line interface (**oc**) as described in Get Started with the CLI.
- You upgraded Red Hat OpenShift AI from version 2.7 or earlier. If you are working in a new installation of Red Hat OpenShift AI, see Adding a CA bundle.

### Procedure

1. Log in to the OpenShift Container Platform as a cluster administrator.
2. Click **Operators → Installed Operators** and then click the Red Hat OpenShift AI Operator.
3. Click the **DSC Initialization** tab.
4. Click the **default-dsci** object.
5. Click the **YAML** tab.
6. Add the following to the **spec** section, setting the **managementState** field to **Managed**:

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: ""
```

7. If you want to use self-signed certificates added to a cluster-wide CA bundle, log in to the OpenShift Container Platform as a cluster administrator and follow the steps as described in Configuring the cluster-wide proxy during installation.
8. If you want to use self-signed certificates in a custom CA bundle that is separate from the cluster-wide bundle, follow these steps:
   a. Add the custom certificate to the **customCABundle** field of the **default-dsci** object, as shown in the following example:

shown in the following example:

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: |
      -----BEGIN CERTIFICATE-----
      examplebundle123
      -----END CERTIFICATE-----
```

b. Click **Save**.
The Red Hat OpenShift AI Operator creates an **odh-trusted-ca-bundle** ConfigMap containing the certificates in all new and existing non-reserved namespaces.

**Verification**

- If you are using a cluster-wide CA bundle, run the following command to verify that all non-reserved namespaces contain the **odh-trusted-ca-bundle** ConfigMap:

```
$ oc get configmaps --all-namespaces -l app.kubernetes.io/part-of=opendatahub-operator |
grep odh-trusted-ca-bundle
```

- If you are using a custom CA bundle, run the following command to verify that a non-reserved namespace contains the **odh-trusted-ca-bundle** ConfigMap and that the ConfigMap contains your **customCABundle** value. In the following command, *example-namespace* is the non-reserved namespace and *examplebundle123* is the customCABundle value.

```
$ oc get configmap odh-trusted-ca-bundle -n example-namespace -o yaml | grep
examplebundle123
```